Employing Attribute-Based Encryption in Systems with Resource Constrained Devices in an Information-Centric Networking Context

Joakim Borgh
Ericsson Research
joakim.borgh@ericsson.com

Edith Ngai
Ericsson Research
Uppsala University
edith.ngai@it.uu.se

Börje Ohlman
Ericsson Research
Borje.Ohlman@ericsson.com

Adeel Mohammad Malik
Ericsson Research
adeel.mohammad.malik@ericsson.com

***Abstract*—Attribute-Based Encryption (ABE) is considered to be one of the most promising ways to be enforce access control in Information-Centric Networking (ICN). As the Internet of Things (IoT) is being considered as one of the primary use cases for ICN it raises the question of the compatibility between IoT and ABE. An important part of the IoT is the resource constrained devices, for them there is a challenge to perform the computationally expensive operations required for ABE.**

In this paper we consider ABE in sensor networks and discuss the strengths and weaknesses of a system solution where the ABE operations are performed on the sensors. To properly discuss these concerns we have implemented two ABE schemes, a Single-authority ABE (SA-CP-ABE) scheme and a Multi-authority ABE (MA-CP-ABE) scheme. Results regarding the execution time, RAM usage, data overhead and battery consumption of these implementations on a sensor are presented. We conclude that it is possible, already today, to perform ABE on sensors for smaller policies. The main limitation in deploying ABE in sensors is the RAM size of the sensors.

I. Introduction

The Internet of Things (IoT) differs from the traditional Internet in a number of ways. This is especially true for the part of the IoT that consists of constrained devices. They can be constrained in a number of ways, e.g. low processing power, limited battery life, lack of end-to-end connectivity, operating in unprotected environments.

To ensure data integrity, privacy and proper access control for these devices is challenging. In this paper we look at to what extent Information-centric Networking (ICN) combined with Attribute Based Encryption (ABE) can provide solution to address these challenges.

Most traditional IoT solutions are based on a concept where the sensors publish sensor readings to the cloud. Users then subscribe to or request the desired data from the cloud services. Communication is protect by end-to-end (e2e) encryption, e.g. TLS or DTLS. In cases where e2e connectivity cannot be established a gateway is needed. These solutions require that the cloud provider and the gateways can be trusted. We believe that there are many scenarios when these assumptions are not true.

978-1-5090-5873-0/17/$31.00

ICN provides a communication paradigm that is based on store and forward communication that gives inherent support for non-e2e communication. Each ICN node can provide caching and can thus serve as a gateway for sleeping and/or unreachable devices. It can also store requests from subscribers until an IoT device wakes up and is capable of responding.

All ICN nodes can thus serve as data caches and relays, but we do not think it is reasonable to require all of them to be trusted devices. Therefore we think object security, where the information object is secured directly on the sensor (data source) is needed.

To fit the needs of ICN, cached objects should be the encrypted version of the objects that can be decrypted by different groups of users with qualified access credentials. ABE allows the use of complex access policies in an encryption scheme that only results in one encrypted version of the object for all access groups. This makes ABE a very efficient scheme for data encryption and dissemination with ICN. The only main drawback with ABE, especially in the context of constrained devices, is that it is computationally heavy. Therefore many people have not really considered ABE to be a viable option in constrained IoT environments. In this paper we show that it is possible to implement ABE and ICN on constrained devices today and that it thereby might become mainstream in the future. By presenting these results, we hope that it can move the discussion forward and drive the community to explore how ABE can assist in making ICN a powerful platform for sensors. We believe that this is our main contribution in this paper.

II. Background

When the current Internet architecture was created, it was designed to connect network nodes e2e. The main use of the Internet today is to retrieve content such as web pages, documents and other types of media. A number of overlay solutions, e.g. CDN and P2P networks, are widely being used to improve the performance of today's Internet. Information-Centric Networking (ICN) attempts to include these very successful techniques into the network layer in an application

978-1-5090-5873-0/17/$31.00 ©2017 IEEE

independent way. In ICN, the content is named not the end-points. Popular ICN architectures are Named Data Networking (NDN) and Content-Centric Networking (CCN) [11].

Communication in CCN uses two packet types: interest packets and content packets. A consumer asks for content by sending an interest packet. Any CCN node which possess the content requested in the interest packet will respond with a content packet with the requested content. A CCN node that cannot satisfy the request itself will forward the interest packet in the direction of the publisher of the content.

ABE is a form of public-key encryption where data is described with attributes as meta-data. The attributes decide how the data is encrypted and only the entities with the corresponding keys should be able to access the content. There are two types of ABE: Key-Policy ABE (KP-ABE) [6] and Ciphertext-Policy ABE (CP-ABE) [4]. In KP-ABE the data is described with attributes and the users' private keys are associated with access policies. A user can decrypt the encrypted data if his/her access policy is satisfied by the attributes from the encrypted data. In CP-ABE the encrypted data, the ciphertext, is associated with an access policy and the users' private keys are associated with attributes. In this scenario the encrypted file can be decrypted if the user's attributes satisfies the file's policy.

ABE systems require a third party, an authority, to setup the public parameters of the system and to generate private keys for the users in the system. In a single-authority (SA) case, the authority must be trusted as it is able to decrypt all the files in the system, since it can generate any private keys. This situation can be circumvented by employing a multiple-authority (MA) system such as [5], [10]. In a well-designed MA-ABE system, the trust is distributed among several authorities. In order to compromise the entire system, multiple malicious authorities are required. MA systems also support the natural division and governance of attributes belonging to different authorities. As an example, consider a document encrypted under the policy `Uppsala University Employee OR Ericsson Employee`, for some collaboration between Ericsson and Uppsala University (UU). In such a case neither Ericsson nor UU might be comfortable with letting a third party issue the attribute of working there. Thus, it would be most suitable if they themselves were authorities governing attributes related to their operations. In this paper we denote the single authority CP-ABE as SA-CP-ABE and the multi-authority CP-ABE as MA-CP-ABE.

ABE offers expressive access control at the expense of computational efficiency. In ABE the time required to perform the encryption and decryption operations scale linearly with the number of attributes in the policy. The operations of ABE are about 100-1000 times more expensive than the RSA operations [3]. This is not an issue for workstations or even smartphones [2] (depending on security strength and complexity of policy), but could a challenge for resource constrained devices such as sensors. One objective of this paper is to study the feasibility of deploying ABE on resource-constrained sensors in the IoT.

III. Related Work

Access control has attracted much attention in information-centric networking. For example, Kurihara et al. [9] proposed an encryption-based access control framework for content-centric networking, called CCN-AC. CCN-AC enables secure content object manifests and supports different access control schemes. It implements two sample access control schemes, including group-based access control and broadcast access control. Ion et al. [7] presented an attribute-based access control in ICNs by applying ABE [4] and proposed a routing scheme based on user's attributes. Most of the existing work on access control for ICN considers servers or computers on the Internet as content producers, which are usually equipped with strong computation capability for data encryption.

With the popularity of smart devices, Internet-of-Things (IoT) is considered as a new paradigm for future Internet. Access control has been considered for securing data objects with information-centric networking for the IoT. In traditional encryption schemes, a sender usually encrypts data that can only be decrypted by an exact recipient. However, in many IoT scenarios, such as smart home and smart cities, senders may not know the identities of all the current and future recipients [14]. ABE [4], [6] is a promising approach that can address these issues in the IoT. It enables expressive and fine-grained data access control policies that built from attributes. An data object can be encrypted one time and be shared by multiple users, which simplifies key management and makes data distribution more efficient. In group-oriented publish-subscribe systems found in the IoT, ABE does not require the group key to be updated whenever a new member joins, which significantly improves the scalability [16].

Several papers have discussed about ABE for the IoT, where ABE operations (e.g. data encryption) are performed at a server or gateway, which the sensor shares a symmetric key with [13]. Although a server or gateway provides stronger computational capabilities than sensors, this system design requires secure communication and trust between the sensor and the server or gateway. In [15], it suggested to delegate the heavy operations of ABE to neighboring unconstrained nodes. However, these neighboring nodes have to be trusted and therefore this is a similar approach to having a server performing all the encryption.

To the best of our knowledge, there is no other paper discussing and showing experimental results of the encryption operation of ABE on a sensor. Recent work has been conducted to explore the feasibility of ABE on smartphones [2], [16]. Nevertheless, the capabilities of smartphones are not as resource constrained as many existing smart devices such as sensors.

Regarding MA-ABE, an important issue is to ensure that users cannot collude and combine their attributes. One solution in MA-ABE is to have a central authority that does not hand out any attributes but only global identifiers (GID). Another issue is whether the attribute authorities would collude. To avoid this problem, there is a proposed solution that protects

security as long as at least two attribute authorities are honest [5]. In this paper we implement the MA-CP-ABE scheme presented in [10] which employs the GID solution.

IV. System Design

A. System overview

We consider an IoT system which connects resource-limited sensors to the Internet via a gateway. In our system, the data gathered by the sensors is considered private and must be encrypted to enforce access control. Only the users with the corresponding access rights should be able to decrypt the data. Our system architecture differs from the existing cloud-based ABE architecture in which the cloud server is utilized to perform heavy computation, such as data encryption.

B. Applying ABE in an IoT context

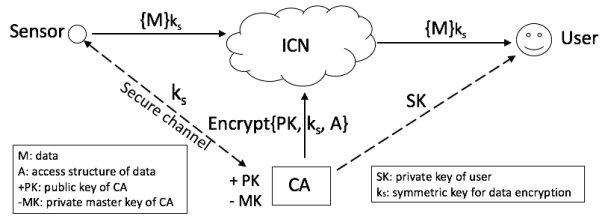
In order to perform ABE on the sensor data, the following sequence of operations is performed.

- Setup: CA generates a public key PK and a private master key MK.
- Encryption: Each sensor uses the public key PK to encrypt its data M following an access structure A by performing $Encrypt(PK, M, A)$, and as a result generating the ciphertext CT. In case that the encryption is too heavy for the sensors, alternative architectures could be employed to offload the ABE encryption from the sensors to the gateway or a server.
- Private key generation: CA will generate the private key (SK) of an individual user based on his set of attributes, S, through $KeyGen(MK, S)$.
- Decryption: a user can decrypt the data by performing $Decrypt(PK, CT, SK)$.

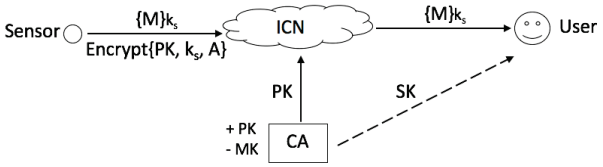
C. Our proposed solutions

We discuss two solutions of how to employ ABE in an IoT system. The two solutions differ in where the ABE operations are applied. In the centralized ABE solution, ABE encryption is offloaded to a server with stronger computation power. This is a common approach in many existing ABE solutions for resource constrained data producers. The ABE sensor system is a novel solution to perform ABE encryption on a sensor, but it is rarely considered due to the questionable feasibility. In this paper we show that the ABE sensor system is indeed feasible for smaller policies.

1) The centralized ABE solution: Figure 1a shows the system architecture and operations in the centralized solution, where ABE is performed at a central authority (CA). In this system, the sensor and the authority share a symmetric key k_s, which can be exchanged via a secure channel. This could be done using TLS, or in a more lightweight fashion by encrypting the symmetric key with the public key of the other party. The sensor encrypts the data M under the symmetric key to obtain $\{M\}_{k_s}$. This encrypted data $\{M\}_{k_s}$ is published to the ICN network, which can be requested by the user. The sensor keeps the symmetric key private. The authority can then proceed to encrypt the symmetric key with ABE under the desired policy and publish this encrypted key to the network. This operation is indicated by $Encrypt\{PK, k_s, A\}$ in the figure, where PK is the public key of the CA and A is the access structure of data M. Note that the published data from the sensor should include an identifier for the symmetric key, so that the user requesting the encrypted data can also request for the encrypted symmetric key. To access the data from the sensors, a user must be able to decrypt the symmetric key successfully and then decrypt the data.



(a) The centralized ABE solution



(b) The ABE sensor system

Fig. 1: Our system designs

The advantage of this system is that the sensors only need to perform symmetric encryption, which is a cheap operation, regardless of the access policy. The biggest drawback of this system is that the authority, which shares the symmetric key with the sensor, will be able to decrypt any message published by the sensor. This system also requires secure end-to-end communication between the sensor and the authority.

2) ABE sensor system: Figure 1b shows the architecture and operations of the ABE sensor system. If the sensors are powerful enough to perform ABE operations, the central authority does not need to be involved in the encryption. The CA is needed only to hand out the private and public keys. It is reasonable to believe that the sensor will be encrypting data under the same policy repeatedly, therefore it is beneficial to encrypt the data using a symmetric key and encrypt this symmetric key with ABE. This way the sensor does not have to perform expensive ABE operations for every data object. Instead the sensor would only need to perform ABE when refreshing the symmetric key used for encryption of the data.

The advantage of this system is that it requires no end-to-end communication between authority and sensor. In addition, if it is a multi-authority ABE system, no single third party will be able to decrypt all the data. The drawback of this ABE system is the computational requirement of the sensors, which limits the feasibility.

In a practical scenario, it is desirable to have multiple authorities for key distribution. In a single authority system, the authority holds a secret key MK that can decrypt all the

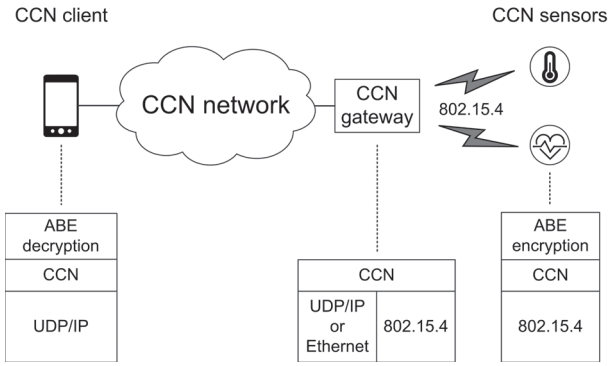
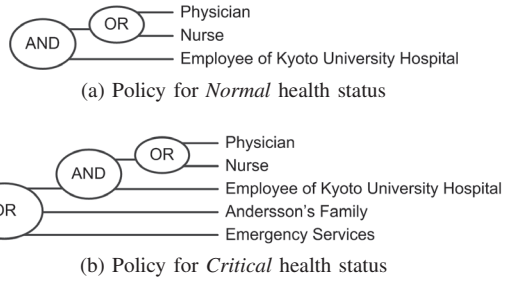


Fig. 2: Implementation setup



(a) Policy for *Normal* health status

(b) Policy for *Critical* health status

Fig. 3: ABE encryption policies

data in the network. With multiple authorities, each authority is responsible for a subset of the attributes and therefore a single authority cannot decrypt all the data. The only exception is when the access control policy can be satisfied by the attributes from a single authority.

D. Key revocation

Key revocation occurs when a sensor (or data publisher) wants to revoke one or multiple users, which is a challenging problem in ABE. When a user's privileges is revoked, the content would have to be re-encrypted and republished. In an ICN context, it becomes even more complicated, since the encrypted data can be cached in any of the routers in the network. It is almost impossible to identify all the cached data and re-encrypt them again.

Proxy re-encryption has been proposed to address the key revocation problem recently [12], [8]. This new scheme supports immediate revocation without the need of re-keying the users or re-encrypting the content. This technique is especially suitable for ICN scenario, where data are cached anywhere in the network. With slight modification, this technique can be adopted into our system. First of all, a semi-trusted proxy will be added to our system. This proxy will not be able to decrypt the encrypted data, but it is able to convert a blinded access structure and make it readable only by the unrevoked users. When a publisher wants to revoke one or multiple users, it forwards the IDs of the revoked users to the proxy.

To facilitate proxy re-encryption, we add in one more component to the encrypted content. The encrypted content now contains two parts, the encrypted data and a blinded access structure. To access the data, the user must send the blinded access structure to the proxy. The proxy then uses its key to transform the blinded access component into a form that the user can combine with his SK to decrypt the data. The proxy key enables the unblinding of the access structure only for the unrevoked users. Hence, the revoked users will not be able to decrypt the data.

V. Implementation

Figure 2 shows the implementation setup that employs the solution described in Section IV-C2. The same setup was adopted in a demo paper [18]. The implementation setup has two sensor devices, a temperature sensor and a heart rate sensor. Data from the temperature sensor is encrypted using single authority ABE, while data from the heart rate sensor is encrypted using multi-authority ABE. The sensors run a CCN stack directly over 802.15.4 radio without any underlying TCP, UDP and IP layers. This makes the communication stack really lightweight leaving more RAM and processor resources for ABE computations.

The SA-CP-ABE is implementation based on the functional encryption library [1], following the scheme presented in [17]. The MA-CP-ABE implementation uses the infrastructure of the functional encryption library to implement the scheme presented in [10], but the scheme has been adjusted to use asymmetric pairing instead of symmetric pairing.

Data is encrypted by the sensors using one of the two ABE policies shown in Figure 3 depending on the health status of the patient. Figure 3a and 3b show the policies when the health status is normal and critical respectively. When the health status is normal, a more restrictive ABE policy is used to encrypt the data. In contrast, a more relaxed policy is used when the health status is critical.

The encrypted data and the ciphertext can be large in ABE. In the case of multi-authority ABE with five attributes, the ciphertext is approximately 3000 bytes. In order to transmit large data over the communication stack, data was chunked down into smaller chunks of 64 bytes where each chunk is a CCN object [19]. This allows fitting the chunk for the CCN and 802.15.4 headers in the 127-byte MTU of 802.15.4.

Since ABE is computationally heavy, it may take a significant amount of time. For example, it takes around 10 seconds to complete multi-authority encryption with five attributes (see Figure 3b). It is not ideal for a CCN client to wait for this long after sending an Interest. The way we handle this is by having an ABE thread that runs in the background to renew the symmetric key, encrypt it with ABE, and store the resulting encrypted symmetric key and associated ciphertext. Upon receiving an Interest, the sensor uses the last generated symmetric key to encrypt the data. This operation is fast, since symmetric key encryption is very lightweight.

We worked with two different sensor platforms over the course of the work done for this paper. Initially we ran ABE on bare metal (without any operating system) and used a sensor platform with a STM32L151VCT6 MCU featuring a

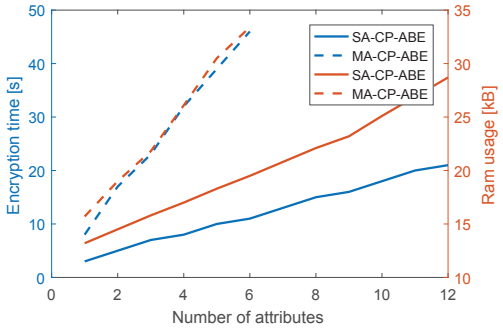


Fig. 4: Encryption time and RAM usage of MA-CP-ABE and SA-CP-ABE on the sensor as a function of number of attributes.

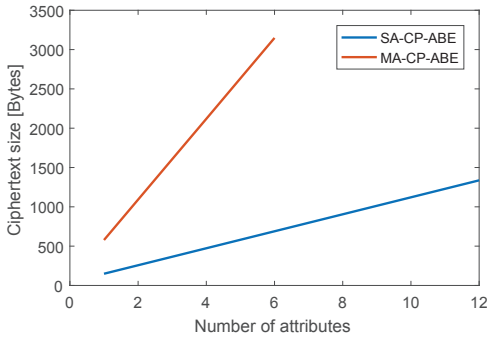


Fig. 5: The size of the ciphertext of MA-CP-ABE and SA-CP-ABE as a function of number of attributes.

256 kB flash and 32 kB RAM. This hardware was also used to make measurements for encryption time, RAM usage and data overhead, which are discussed in section VI-A. This ABE implementation was later ported to a more capable hardware platform, i.e. STM32F4DISCOVERY running the RIOT OS [20] and CCN-lite [21] for the communication stack. Figure 2 shows the final setup where STM32F4DISCOVERY was used for both the sensors and the CCN gateway. This platform has a STM32F407 MCU that features an ARM Cortex-M4 32-bit core, 1 MB Flash memory and 192 kB RAM. Both the sensor and the gateway run the RIOT OS [20] and CCN-lite [21] in our implementation.

VI. Results & Discussion

To investigate the feasibility of ABE on constrained sensor devices, experimental measurements for encryption time, RAM usage, data overhead and battery consumption of ABE were carried out. The measurements for encryption time, RAM usage and data overhead were performed on a sensor platform with the STM32L151VCT6 MCU bare metal, whereas the measurements for battery consumption were conducted on the STM32F4DISCOVERY platform with the STM32F407 MCU running the RIOT OS.

A. Encryption time, RAM usage and ciphertext size

Both the SA-CP-ABE and MA-CP-ABE implementation use the 256-bit pairing friendly curve of the relic-toolkit that yields a 128-bit security level. Figure 4 shows the encryption time and the RAM usage varying the number of attributes for MA-CP-ABE and SA-CP-ABE. The policy size ranges from 1 to 6 attributes for MA-CP-ABE and 1 to 12 attributes for SA-CP-ABE. The encryption time and the RAM usage scale linearly with number of attributes, which is an expected result for ABE. From the figure, MA-CP-ABE is significantly slower than SA-CP-ABE. Every added attribute in MA-CP-ABE adds approximately 7 seconds whereas an added attribute in SA-CP-ABE adds slightly less than 2 seconds. SA-CP-ABE consumes substantially less RAM than MA-CP-ABE. The upper limits for the number of attributes are set because the sensor does not have sufficient RAM to support larger policies.

ABE depends on the pairing operation, which is a bilinear map, e, between two group elements (of potentially the same group) to an element of a third group. Typically denoted by:

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T.$$

The relevance of this is that the workload of the encryption algorithm in ABE varies heavily depending on in which group the operation is applied. In Table I the execution time of the main cryptographic operations of the encryption algorithm of ABE on both a laptop and the sensor can be seen. The complexity of the SA-CP-ABE encryption operation is 1 exponentiation in $\mathbb{G}_T$, $2n$ exponentiations in $\mathbb{G}_1$ and n exponentiations in $\mathbb{G}_2$, where n is the number of attributes in the policy. The complexity of the MA-CP-ABE encryption operation is $2n+1$ exponentiations in $\mathbb{G}_T$ and $3n$ exponentiations in $\mathbb{G}_2$. Table I shows the execution time of the exponentiation operations on a laptop and a sensor. With the knowledge of these complexities and the data presented in Table I, one can clearly understand why MA-CP-ABE encryption is much slower than SA-CP-ABE encryption.

The ciphertext size of the two schemes can be seen in Figure 5. Once again, we can see that SA-CP-ABE is significantly more lightweight than MA-CP-ABE. The ciphertext in SA-CP-ABE consists of $n+1$ $\mathbb{G}_1$ elements, n $\mathbb{G}_2$ elements and the policy string. In MA-CP-ABE, the ciphertext contains $2n$ $\mathbb{G}_2$ elements, n $\mathbb{G}_T$ elements and the policy string, where n is the number of attributes in the policy. The main reasons of why the ciphertext of MA-CP-ABE is so much larger than that of SA-CP-ABE is because $\mathbb{G}_1$ and $\mathbb{G}_2$ elements can be compressed in the relic-toolkit, but unfortunately the $\mathbb{G}_T$ elements can not be compressed. Additionally, the $\mathbb{G}_2$ elements are approximately twice the size of $\mathbb{G}_1$ elements and $\mathbb{G}_T$ elements are approximately six times the size of $\mathbb{G}_1$ elements. The size of these elements, and therefore the size of the ciphertext as well, are dependent on the security level.

B. Battery consumption

To better understand the battery requirements of ABE, we carried out some measurements on battery consumption of the ABE thread in the STM32F4DISCOVERY sensor platform. The measurements were carried out for a thread that runs

TABLE I: The execution time of the main cryptographic operations on a laptop and the sensor.

Operation	Time on laptop [s]	Time on sensor [s]
exp. in $\mathbb{G}_1$	$1.2 \cdot 10^{-3}$	0.22
exp. in $\mathbb{G}_2$	$3.6 \cdot 10^{-3}$	1.16
exp. in $\mathbb{G}_T$	$1.1 \cdot 10^{-2}$	2.05

multi-authority ABE with five attributes over a period of 30 minutes. During this period, the sensor device managed to encrypt a 16 byte symmetric key 165 times with the total battery consumed of 14.53 mAh. Therefore, one multi-authority encryption with five attributes consumes battery capacity equivalent to 0.088 mAh.

Assuming the case that the symmetric key is renewed once per day, the battery consumed by the ABE computations in one day will be around 0.088 mAh. A typical AA battery, with a capacity of 2000 mAh, will be depleted in 22705 days or approximately 62 years. Here we disregard the energy consumed by the hardware and other software threads such as the communication stack. We only account for energy consumed by the ABE thread and running the RIOT OS.

Another aspect that should be taken into account is the energy required to transmit the ciphertext. Multi-authority ABE can generate ciphertext of approximately 3 kB with five attributes. However, the ciphertext does not need to be transmitted with every sensor reading. It only needs to be transmitted once when the symmetric key is renewed.

VII. Conclusions and Future work

We have implemented two ABE schemes, SA-CP-ABE and MA-CP-ABE, in C which can be run on resource constrained sensors. Experimental results of the execution time, RAM usage, data overhead and battery consumption of the encryption operation of these schemes are presented along with discussion regarding feasibility and possible improvements of the implementation.

The resource consumption of ABE computations depends on a number of parameters such as the frequency of ABE encryption, size of the data encrypted, encryption mode used (single or multi-authority), number of attributes, etc. We argue that the feasibility of ABE on constrained devices depends on the use case and the security requirements of the application. We demonstrated that ABE can certainly be run on constrained devices as proposed in Section IV-C2 with sufficiently large ABE policies to address most use cases. The major limiting factor of the feasibility of performing ABE on resource constrained devices is their RAM size. The encryption time issue can be circumvented potentially by using the same session key for a certain amount of time and refreshing it periodically. The MA-CP-ABE implementation is significantly more resource demanding than the SA-CP-ABE implementation.

By showing the feasibility of deploying ABE in a constrained IoT system, we also show that ABE can be used as a general access control mechanism in an ICN context from the smallest devices to large cloud and media distribution systems. The gain of performing ABE operations on the sensors is largest in a multiple authority scenario as this removes the single trusted third party.

Our implementation has room for improvements. In particular using a library for the sensor which utilizes built-in hardware support for some cryptographic operations should reduce the time needed for encryption. In the future, we would also like to add features to make this system more dynamic e.g., installing new ABE policies in the sensor and enabling the data requester to specify the ABE policy that the sensor should use to encrypt the data.

References

[1] Functional Encryption Library, https://code.google.com/archive/p/libfenc/

[2] M. Ambrosin, M. Conti, T. Dargahi, *On the Feasibility of Attribute-Based Encryption on Smartphone Devices*, arXiv:1504.00619.

[3] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, D. Starin, *Persona: An online social network with user-defined privacy*, SIGCOMM Computer Communications Review, 39(4):135–146, 2009.

[4] J. Bethencourt, A. Sahai, B. Waters, *Ciphertext-Policy Attribute-Based Encryption*, In IEEE Symposium on Security and Privacy, 2007, pp. 321-334.

[5] M. Chase, S.M. Chow, *Improving privacy and security in multi-authority attribute-based encryption*, In ACM conference on Computer and communications security (CCS), 2009, pp. 121–130.

[6] V. Goyal, O. Pandey, A. Sahai, B. Waters, *Attribute Based Encryption for Fine-Grained Access Conrol of Encrypted Data*, In ACM conference on Computer and Communications Security (CCS), 2006.

[7] M. Ion, J. Zhang, and E. M. Schooler, *Toward content-centric privacy in ICN: Attribute-based encryption and routing*, In ACM SIGCOMM workshop on Information-centric networking, 2013.

[8] Sonia Jahid, and Prateek Mittal, and Nikita Borisov, *EASiER: encryption-based access control in social networks with efficient revocation*, In ACM Symposium on Information, Computer and Communications Security (ASIACCS), 2011.

[9] J. Kuriharay, E. Uzun, and C. A. Wood, *An encryption-based access control framework for content-centric networking*, In IFIP Networking Conference, 2015.

[10] A. Lewko, B. Waters, *Decentralizing Attribute-Based Encryption*, Cryptology ePrint Archive Report 2010/351 (2010), http://eprint.iacr.org/

[11] P. Mahadevan, *CCNx 1.0 Tutorial*, PARC, Tech. Rep., March 2014.

[12] R. S. da Silva, and S. D. Zorzo, *An access control mechanism to ensure privacy in named data networking using attribute-based encryption with immediate revocation of privileges*, In Annual IEEE Consumer Communications and Networking Conference (CCNC), 2015.

[13] Y-L. Tan, B-M. Goi, R. Komiya, S-Y. Tan, *A Study of Attribute-Based Encryption for Body Sensor Networks*, Informatics Engineering and Information Science (2011): 238-247.

[14] D. Thatmann, S. Zickau, A. Förster, A. Küpper, *Applying Attribute-Based Encryption on Publish Subscribe Messaging Patterns for the Internet of Things*, In IEEE International Conference on Data Science and Data Intensive Systems, 2015, pp. 556-563.

[15] L. Touati, Y. Challal, A. Bouabdallah, *C-CP-ABE: Cooperative Ciphertext Policy Attribute-Based Encryption for the Internet of Things*, In International Conference on advanced Networking, Distributed Systems and Applications, 2014, pp. 64-69.

[16] X. Wang, J. Zhang, E. M. Schooler, M. Ion. *Performance evaluation of attribute-based encryption: Toward data privacy in the IoT*, In IEEE International Conference on Communications (ICC), 2014.

[17] B. Waters, *Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization*, Cryptology ePrint Archive Report 2008/290 (2008), http://eprint.iacr.org/

[18] A. M. Malik, J. Borgh, B. Ohlman, *Attribute-Based Encryption on a Resource Constrained Sensor in an Information-Centric Network*, In ACM Conference on Information-Centric Networking, 2016, pp. 217-218.

[19] M. Mosko, *CCNx Content Object Chunking*, draft-mosko-icnrg-ccnxchunking-02.

[20] RIOT OS for the Internet-of-Things, https://www.riot-os.org/

[21] CCN-lite: Lightweight implementation of the CCNx protocol of XEROX PARC, http://www.ccn-lite.net/